

Algoritmos Híbridos para el Problema de Scheduling de Máquinas Paralelas

Gatica Claudia Ruth, Esquivel Susana Cecilia

Laboratorio de Investigación y Desarrollo en Inteligencia Computacional (LIDIC)

Departamento de Informática

Universidad Nacional de San Luis

Ejército de los Andes 950 - Local 106

(5700) - San Luis - Argentina

Tel: (02652) 420823 / Fax: (02652) 430224

cratica,esquivel@unsl.edu.ar

Resumen

En los últimos años ha surgido un gran interés en las metaheurísticas híbridas, las cuales han sido desarrolladas considerablemente en el campo de la optimización. Combinaciones de algoritmos han provisto técnicas muy poderosas de algoritmos de búsqueda. En la literatura podemos citar los siguientes trabajos en relación a las metaheurísticas híbridas: [1], [6], [4], [8] y [10]. En esta etapa de nuestro trabajo hemos estudiado una variante de Algoritmos Genéticos (AGs) que pertenece a la clase de metaheurísticas poblacionales (*P*-metaheurísticas) [2] y su hibridización con dos metaheurísticas de solución única (*S*-metaheurísticas) son: Recocido Simulado (SA) y Búsqueda Local (LS). El problema en estudio es minimizar la máxima tardanza (*Maximum Tardiness*) en scheduling de máquinas paralelas independientes. La

notación usada en la literatura [7] para describir el mismo es una triupla: $(\alpha|\beta|\gamma)$. El primer campo, α describe el ambiente de máquinas, el segundo campo, β indica las restricciones entre las tareas que son asignadas a las máquinas. Por último, el tercer campo, γ provee la o las funciones objetivo a ser optimizadas. Acorde a esta notación, nuestro problema se describe mediante $(P_m || T_{max})$. Tal problema se ha considerado en [9] de complejidad NP-duro, cuando $2 \leq m \leq n$ (m es el número de máquinas y n el número de tareas). Los resultados obtenidos hasta ahora con la propuesta de un AG híbrido han mostrado una mejora considerable en la minimización de la máxima tardanza, esto se observa principalmente en las instancias de mayor complejidad.

Palabras claves: Algoritmos Híbridos, Scheduling de Máquinas paralelas, Máxima Tardanza, Algoritmos Genéticos, Algoritmo

Recocido Simulado, Búsqueda Local.

1. Contexto

La línea de investigación se enmarca en el paradigma de “Inteligencia Colectiva”, la cual está inserta en el proyecto “Metaheurísticas Avanzadas para Problemas de Optimización con y sin Restricciones”. Proyecto en ejecución desde Enero de 2010 e inserto dentro del Laboratorio de Investigación y Desarrollo en Inteligencia Computacional (LIDIC).

2. Introducción

Entre las clasificaciones de algoritmos híbridos existentes se pueden encontrar esquemas clasificación plana y jerárquica. La taxonomía por nosotros adoptada es la propuesta en [2], [3]. La hibridización de metaheurísticas involucra algunas cuestiones principales que pueden ser clasificadas como de diseño e implementación. En cuestión de diseño la clasificación jerárquica involucra las hibridizaciones de bajo nivel (*low level*) y de alto nivel (*high level*). En la hibridización de bajo nivel, una función de una metaheurística dada es reemplazada por alguna otra metaheurística. En algoritmos híbridos de alto nivel, las diferentes metaheurísticas son autocontenidas, no hay una relación directa en el trabajo interno de una metaheurística y otra. En la clasificación plana distinguimos la hibridización de relevo (*relay*) y la cooperativa (*teamwork*). En la hibridización de relevo, las metaheurísticas son ejecutadas una después de la otra, cada una usa la sal-

ida de la metaheurística previa actuando en una forma de tubería (*pipeline*). La hibridización de cooperación (*teamwork*) representa modelos de optimización cooperativa, en los cuales muchos agentes evolucionan en paralelo. Finalmente encontramos combinaciones entre las clasificaciones jerárquica y plana. En esta etapa de nuestro trabajo hemos estudiado un *AG* y tres formas posibles de hibridización, combinando su potencial de búsqueda explorativa con los algoritmos de solución única *SA* y *LS*, los cuales realizan una búsqueda explotativa. El primer algoritmo híbrido construido es $H_1 : AG + SA + LS$, el cual pertenece a un modelo de hibridización de alto nivel y relevo (*HRH*). Esto porque cada algoritmo es autocontenido y al finalizar cada uno de ellos comienza el otro, más detalladamente, cuando *AG* completa su ciclo evolutivo, la mejor solución encontrada hasta el momento es usada por *SA* como su solución inicial, cuando *SA* termina su ciclo evolutivo, la solución actual de *SA* es usada por *LS* como su solución inicial. El segundo algoritmo híbrido desarrollado es $H_2 : AG + SA + LS$, este es un modelo de bajo nivel y relevo (*LRH*), en el cual el algoritmo *SA* es invocado como un operador de mejoramiento de la mejor solución actual dentro del ciclo evolutivo de *AG*. Cuando *AG* finaliza su ciclo evolutivo, se invoca al algoritmo *LS*. El tercer algoritmo híbrido desarrollado $H_3 : AG + SA + LS$, es también un modelo híbrido de bajo nivel y relevo (*LRH*). A diferencia de H_2 , en H_3 se elige un individuo al azar desde la población inicial de *AG* y éste es mejorado por *SA*, luego es nuevamente insertado en la población inicial de *AG*, cuando *AG* termina su ciclo, *LS* es invocado. Los experimentos realizados y

los resultados obtenidos son mostrados más adelante.

3. Líneas de investigación y desarrollo

La línea de investigación se encuentra en el paradigma de “Inteligencias Colectivas”, los algoritmos que se están desarrollando son dos P-metaheurísticas (algoritmos poblacionales), estos son los Algoritmos Genéticos (AG) y los algoritmos de Colonias de Hormigas (ACO) y sus posibles hibridizaciones con las S-metaheurísticas (algoritmos de única solución), por ejemplo, el Recocido Simulado (SA) y la Búsqueda Local (LS).

SA es una técnica basada en la analogía del proceso de enfriamiento (annealing) de los sólidos, donde un sólido es calentado a una temperatura alta y gradualmente es enfriado en orden a ser cristalizado. SA realiza el proceso de búsqueda local de la siguiente manera: A partir de la solución inicial s_0 se genera una la solución s_1 a través de un operador de movimiento, s_1 es aceptada como la nueva solución actual si δ es menor que cero. Donde $\delta = f(s_1) - f(s_0)$. Para permitir que la búsqueda local escape de un óptimo local, los movimientos que incrementan los valores de la función objetivo son aceptados con una probabilidad $\exp(-\delta/T)$ si δ es mayor que cero, T es un parámetro llamado “Temperatura”.

LS (en la literatura nos referimos a la S-metaheurística en general) es el más antiguo y simple método [2]. Este comienza con una solución inicial dada. En cada iteración la metaheurística reemplaza la solución actual

por un vecino que mejora la función objetivo. La búsqueda se detiene cuando todos los vecinos candidatos son peores que la solución actual, lo que significa que se ha alcanzado un óptimo local. El principal objetivo del vecindario es acelerar la búsqueda. Las variantes de LS pueden ser distinguidas acorde a cómo son generadas y a la estrategia de selección de las soluciones desde el vecindario.

4. Resultados y Objetivos

En la mayoría de los casos de optimización, los algoritmos de búsqueda requieren la determinación de los parámetros antes que la optimización en sí sea realizada. Para elegir los parámetros de los algoritmos usados en este trabajo, se consideraron los parámetros específicos de los algoritmos involucrados y se estudiaron separadamente mediante una experimentación previa, mediante el diseño factorial descripto en [11]. Las instancias del problema usadas en los experimentos fueron 20 instancias obtenidas de la *OR-Library* [5], de tamaño de 100 tareas (*jobs*) y 5 máquinas. Las cuales tienen un factor de tardanza que va en incremento de acuerdo con el índice de la instancia. Para simplificar el análisis de los resultados se dividió al conjunto de instancias en 2 grupos: el grupo 1 esta compuesto por las 10 primeras instancias (de menor complejidad), y el grupo 2 involucra a las restantes 10 instancias (de mayor complejidad).

La función objetivo es minimizar la máxima tardanza (T_{max}). Para lograr una com-

paración justa del algoritmo AG y sus versiones híbridas se estableció un número máximo de evaluaciones para todos los algoritmos involucrados: AG , H_1 , H_2 y H_3 .

Los parámetros usados para AG fueron: cantidad de corridas 30, operador de crossover OX (con probabilidad 0,90) y operador de mutación n -swaps (con probabilidad 0,30).

Los parámetros usados para los híbridos H_1 , H_2 y H_3 fueron para AG : operador de crossover OX (con probabilidad 0,90) y operador de mutación n -swaps (con probabilidad 0,30). Para SA : temperatura inicial 10000, razón de enfriamiento 0,90, longitud de la cadena de Markov 1000 y operador de movimiento $swap$. Para LS : tamaños de vecindario 30, operadores de movimientos: $swap$ y $scramble$. Las figuras 1 y 2 ilustran los diagramas de cuadro del *ErrorPorcentual* de T_{max} para grupo 1 y 2, respectivamente cuando LS usa el operador de movimiento $swap$ (este mismo operador es usado en el algoritmo SA). Similarmente, las figuras 3 y 4 muestran los diagramas de cuadro del *ErrorPorcentual* de T_{max} grupo 1 y 2, cuando LS usa el operador de movimiento $scramble$.

Analizando el grupo 1, en las figuras 1 y 3, ilustran los diagramas de cuadro de los algoritmos, se observa que AG alcanza valores cercanos a cero, pero H_1 , H_2 y H_3 logran valores mejores y menores a cero, es decir que las versiones híbridas muestran un comportamiento bueno con respecto al *ErrorPorcentual* de T_{max} . Mayormente, se observa en las figuras 2 y 4, las cuales ilustran los diagramas de cuadro de los algoritmos y el *ErrorPorcentual* de T_{max} del grupo 2, una marcada diferencia, se puede notar que el diagrama de cuadro del algo-

ritmo AG esta entre valores mayores a cero, y los diagramas de cuadro de las versiones híbridas son menores y cercanos a cero. En este trabajo no analizaremos las versiones híbridas entre sí, sólo dejamos indicado mediante los resultados obtenidos que los esquemas híbridos han mejorado los resultados que se han obtenido con el algoritmo AG . Estos resultados nos alientan en continuar con el desarrollo de la hibridización de otras P-metaheurísticas en estudio.

5. Formación de Recursos Humanos

El presente trabajo corresponde al desarrollo de la tesis doctoral: “*Desarrollo y Aplicación de metaheurísticas basadas en Inteligencia Computacional para resolver problemas de Planificación de Máquinas Paralelas con un enfoque mono y Multiobjetivo*”, de la Carrera de Posgrado: *Doctorado en Ciencias de la Computación*. El trabajo se divide en dos etapas principales. La Etapa I que involucra el enfoque mono objetivo del desarrollo y la aplicación de las heurísticas basadas en Inteligencia Computacional para la resolución del problema de planificación de máquinas paralelas irrestricto. La Etapa II comprende el enfoque multiobjetivo del desarrollo y la aplicación de la metaheurística que mejor se haya desempeñado en la Etapa I.

Referencias

- [1] A. Roach and N. Rakesh, “A hybrid GA-SA algorithm for just-in-time scheduling of multi-level assemblies”.

- Computers & Industrial Engineering
Volume 30, Issue 4, September 1996,
Pages 1047-1060.
- [2] E.G. Talbi, "Metaheuristics from design to implementation", by John Wiley & Sons, Canada, 2009.
- [3] E.G. Talbi, "A Taxonomy of Hybrid Metaheuristics". Journal of Heuristics Volume 8, Number 5, 541-564, DOI: 10.1023/A:1016540724870.
- [4] H. Zhou and Y. Feng and L. Han, "The hybrid heuristic genetic algorithm for job shop scheduling". Computers & Industrial Engineering Volume 40, Issue 3, July 2001, Pages 191-200.
- [5] J. Beasley, OR-Library. "<http://people.brunel.ac.uk/mastjjb/info.html>".
- [6] K.P. Dahal and G.M. Burt and J.R. McDonald and S.J. Galloway, "GA/SA-based hybrid techniques for the scheduling of generator maintenance in power systems". Evolutionary Computation, 2000. Proceedings of the 2000 Congress on 2000, 567 - 574 vol.1, 16 jul 2000. La Jolla, CA , USA. ISBN: 0-7803-6375-2.
- [7] L.R. Graham and E.R. Lawler and J.K. Lestra and A.A.G. Rinnooy Kan, "Optimization and approximation in deterministic sequencing and scheduling: a survey", Ann Discrete Math, 287-326, 1979.
- [8] L. Wang and D. Zheng, "An effective hybrid optimization strategy for job-shop scheduling problems". Computers & Operations Research 28 (2001) 585-596.
- [9] M. Pinedo, "Scheduling: Theory, Algorithms and System", Prentice Hall, 1995.
- [10] S. K. Ong and J. Ding; A. Y. C. Nee , "Hybrid GA and SA dynamic set-up planning optimization". International Journal of Production Research Volume 40, Issue 18, 2002, Pages 4697 - 4719.
- [11] T. Bartz-Beielstein, "Experimental Research in Evolutionary Computation", The New Experimentalism, Springer, 2006.
- [12] T. Morton and D. Pentico, "Heuristic Scheduling Systems, John Wiley and Sons", 1993, New York.

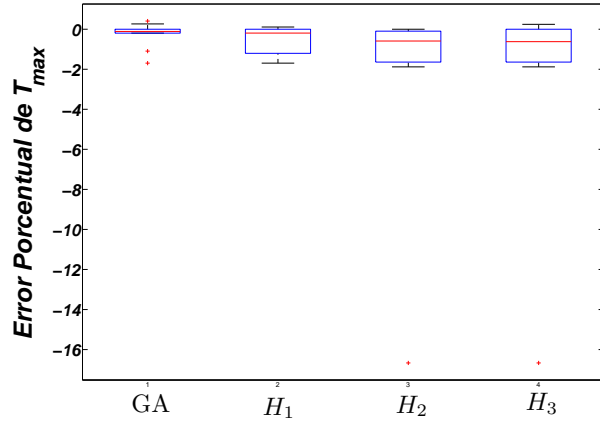


Figura 1: Error Porcentual de T_{max} del grupo 1 y el operador de LS es *swap*.

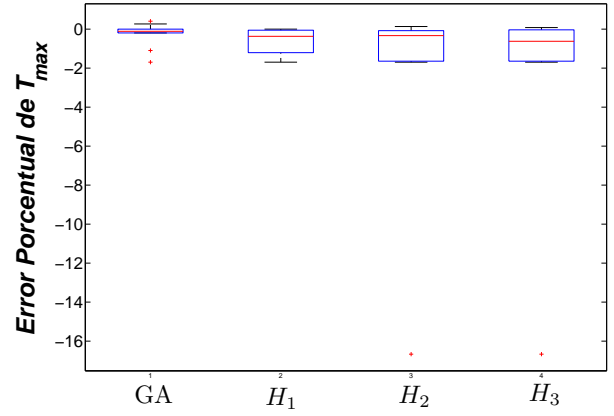


Figura 3: Error Porcentual de T_{max} del grupo 1 y el operador de LS es *scramble*.

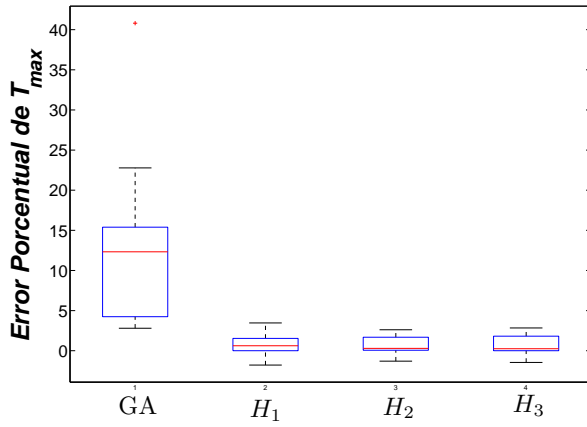


Figura 2: Error Porcentual de T_{max} del grupo 2 y el operador de LS es *swap*.

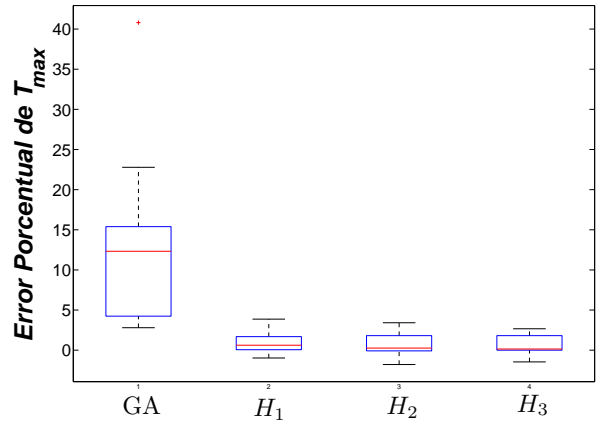


Figura 4: Error Porcentual de T_{max} del grupo 2 y el operador de LS es *scramble*.